Published in "New Journal of Physics 11: 123008, 2009"
which should be cited to refer to this work.

Accurate and diverse recommendations via eliminating redundant correlations

Tao Zhou[1,2], Ri-Qi Su[1], Run-Ran Liu[1], Luo-Luo Jiang[1], Bing-Hong Wang[1,3,4] and Yi-Cheng Zhang[2,3]

[1] Department of Modern Physics and Nonlinear Science Center, University of Science and Technology of China, Hefei Anhui 230026, People's Republic of China

[2] Department of Physics, University of Fribourg, Chemin du Musée 3, CH-1700 Fribourg, Switzerland

[3] Research Center for Complex System Science, University of Shanghai for Science and Technology, Shanghai 200093, People's Republic of China

E-mail: bhwang@ustc.edu.cn

Abstract. In this paper, based on a weighted projection of a bipartite user-object network, we introduce a personalized recommendation algorithm, called *network-based inference* (NBI), which has higher accuracy than the classical algorithm, namely *collaborative filtering*. In NBI, the correlation resulting from a specific attribute may be repeatedly counted in the cumulative recommendations from different objects. By considering the higher order correlations, we design an improved algorithm that can, to some extent, eliminate the redundant correlations. We test our algorithm on two benchmark data sets, *MovieLens* and *Netflix*. Compared with NBI, the algorithmic accuracy, measured by the ranking score, can be further improved by 23% for *MovieLens* and 22% for *Netflix*. The present algorithm can even outperform the *Latent Dirichlet Allocation* algorithm, which requires much longer computational time. Furthermore, most previous studies considered the algorithmic accuracy only; in this paper, we argue that the diversity and popularity, as two significant criteria of algorithmic performance, should also be taken into account. With more or less the same accuracy, an algorithm giving higher diversity and lower popularity is more favorable. Numerical results show that the present algorithm can outperform the standard one simultaneously in all five adopted metrics: lower

[4] Author to whom any correspondence should be addressed.

ranking score and higher precision for accuracy, larger Hamming distance and lower intra-similarity for diversity, as well as smaller average degree for popularity.

Contents

1. Introduction

The exponential growth of the Internet [1] and World Wide Web [2] confronts people with information overload: they encounter too much data and sources to be able to find those most relevant for them. People may choose from thousands of movies, millions of books and billions of web pages. The amount of information is increasing more quickly than our processing ability, with the result that evaluating all these alternatives and then making a choice becomes infeasible. A landmark for information filtering is the use of search engines [3, 4], by which users can find the relevant webpages with the help of properly chosen keywords. However, the search engine has two essential disadvantages. On the one hand, it does not take into account personalization and thus returns the same results for people with far different habits. So, if a user's habits are different from the mainstream, even with some 'right keywords', it is hard for him to filter out what he likes from the countless search results. On the other hand, some tastes, for example musical and poetic, cannot be expressed by keywords, or even language strings. The search engine, based on text matching, will lose its effectiveness in those cases.

Thus far, the most promising way to efficiently filter out the information overload is to provide personalized recommendations. That is to say, using the personal information of a user (i.e. the historical track of this user's activities and possibly her/his personal profile) to uncover his habits and to consider them in the recommendation. For example, *Amazon.com* uses one's purchase record to recommend books [5], *AdaptiveInfo.com* uses one's reading history to recommend news [6], and the *TiVo* digital video system recommends TV shows and movies on the basis of users' viewing patterns and ratings [7].

Motivated by the significance for economy and society [8], the design of an efficient recommendation algorithm becomes a joint focus from engineering science to marketing practice, from mathematical analysis to the physics community (see the review article [9] and the references therein). Various kinds of algorithms have been proposed, including collaborative filtering (CF) [10], content-based analysis [11], spectral analysis [12], iteratively self-consistent refinement [13], principal component analysis [14] and so on.

Very recently some physical dynamics, including the heat conduction process [15] and mass/energy diffusion [16]–[18], have found applications in personalized recommendation. These physical approaches have been demonstrated to be both highly efficient and of

low computational complexity. In this paper, we will first introduce a network-based recommendation algorithm, called *network-based inference* (NBI) [16], which has higher accuracy than the classical CF algorithm. In NBI, the correlation resulting from a specific attribute may be repeatedly counted in the cumulative recommendations from different objects. By considering the higher order correlations, we next design a higher effective algorithm that can, to some extent, eliminate the redundant correlations. Numerical results demonstrate that the improved algorithm has much higher accuracy.

In addition, we here argue that the most accurate recommendations may not be the most useful ones since the more important value added by a recommendation system is to help users to find results that they are unlikely to discover by themselves; namely diversity and novelty should be taken into consideration. Despite this fact, most previous algorithms focus overwhelmingly on accuracy (mostly the accuracy metrics are the only measurements used to evaluate the algorithms [10], and the Netflix Prize [19] challenged researchers to increase the accuracy without any reference to diversity and novelty). We here test the algorithms according to three different metrics: two for diversity and one for novelty. The results indicate that the improved algorithm not only largely enhances the accuracy, but can also provide more diverse and novel recommendations.

2. NBI for personal recommendation

A recommendation system consists of users and objects, and each user has collected some objects. Denoting the object-set as $O = \{o_1, o_2, \ldots, o_n\}$ and user-set as $U = \{u_1, u_2, \ldots, u_m\}$, the recommendation system can be fully described by a bipartite network with $n+m$ nodes, where an object is connected with a user if and only if this object has been collected by this user. Connections between two users or two objects are not allowed. Based on the bipartite user–object network, an object–object network can be constructed, where each node represents an object, and two objects are connected if and only if they have been collected simultaneously by at least one user. We assume a certain amount of resource (i.e. recommendation power) is associated with each object, and the weight w_{ij} represents the proportion of the resource o_j would like to distribute to o_i. For example, in the book-selling system, the weight w_{ij} contributes to the strength of recommending the book o_i to a customer provided he has already bought the book o_j.

The weight w_{ij} can be determined following a network-based resource-allocation process [20] where each object distributes its initial resource equally to all the users who have collected it, and then each user sends back what he has received equally to all the objects he has collected. Figure 1 gives a simple example, where the three X-nodes are initially assigned weights x, y and z. The resource-allocation process consists of two steps; first from X to Y, then back to X. The amount of resource after each step is marked in figures 1(b) and (c), respectively. Merging these two steps into one, the final resource located in the three X-nodes, denoted by x', y' and z', can be obtained as

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 11/18 & 1/6 & 5/18 \\ 1/9 & 5/12 & 5/18 \\ 5/18 & 5/12 & 4/9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{1}$$

According to the above description, this 3×3 matrix is the particular weighted matrix we want. Clearly, this weighted matrix, equivalent to a weighted projection network of X-nodes,

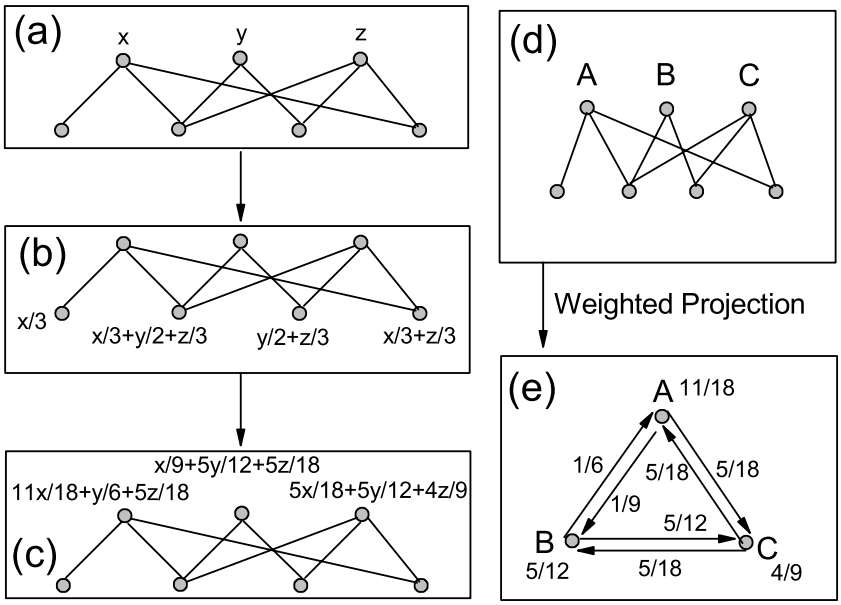

Figure 1. Illustration of the resource-allocation process in a bipartite network. In plots (a)–(c), the upper three are X-nodes, and the lower four are Y-nodes. The whole process consists of two steps: First, the resource flows from X to Y (a→b), and then returns to X (b→c). The process from (a) to (c) can be considered as a weighted projection of a bipartite network, shown as (d)→(e). The weight located on the directed edge A→B means the fraction of resource node A would transfer to node B. The weights of self-connections are also labeled besides the corresponding nodes.

is independent of the initial resources assigned to the X-nodes. A network representation is shown in figures 1(d) and (e). For a general user–object network, the weighted projection onto an object–object network reads [16]

$$w_{ij} = \frac{1}{k(o_j)} \sum_{l=1}^{m} \frac{a_{il} a_{jl}}{k(u_l)}, \tag{2}$$

where $k(o_j) = \sum_{i=1}^{m} a_{ji}$ and $k(u_l) = \sum_{i=1}^{n} a_{il}$ denote the degrees of object o_j and user u_l, and $\{a_{il}\}$ is an $n \times m$ adjacent matrix of the bipartite user–object network, defined as

$$a_{il} = \begin{cases} 1, & o_i \text{ is collected by } u_l, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For a given user u_i, we assign some resource (i.e. recommendation power) on those objects already collected by u_i. In the simplest case, the initial resource vector $\mathbf{f}$ can be set as

$$f_j = a_{ji}. \tag{4}$$

That is to say, if the object o_j has been collected by u_i, then its initial resource is unity, otherwise it is zero. After the resource-allocation process, the final resource vector is

$$\mathbf{f}' = W\mathbf{f}. \tag{5}$$

Accordingly, all u_i's uncollected objects o_j ($1 \leqslant j \leqslant n$, $a_{ji} = 0$) are sorted in descending order of f'_j, and those objects with the highest values of final resource are recommended. We call this method NBI, since it is based on the weighted object–object network [16].

For comparison, we briefly introduce two classical recommendation algorithms. The first is the so-called *global ranking method* (GRM), which sorts all the objects in descending order of degree and recommends those with the highest degrees. The second is the most widely applied recommendation algorithm, named CF [10]. This algorithm is based on measuring the similarity between users or objects. The most widely used similarity measure, also adopted in this paper, is the so-called *Sørensen index* (i.e. the cosine similarity) [21]. For two users u_i and u_j, their cosine similarity is defined as (for more local similarity indices as well as the comparison of them, see [22, 23]):

$$s_{ij} = \frac{1}{\sqrt{k(u_i)k(u_j)}} \sum_{l=1}^{n} a_{li} a_{lj}. \tag{6}$$

For any user–object pair $u_i - o_j$, if u_i has not yet collected o_j (i.e. $a_{ji} = 0$), the predicted score, v_{ij} (to what extent u_i likes o_j), is given as

$$v_{ij} = \frac{\sum_{l=1,l\neq i}^{m} s_{li} a_{jl}}{\sum_{l=1,l\neq i}^{m} s_{li}}. \tag{7}$$

For any user u_i, all the nonzero v_{ij} with $a_{ji} = 0$ are sorted in descending order, and those objects at the top are recommended. This algorithm is based on the similarity between user pairs; we therefore call it user-based CF, abbreviated as UCF. The main idea embedded in UCF is that the target user will be recommended the objects collected by those users sharing similar tastes. Analogously, the recommendation list can be obtained by object-based CF (OCF), that is, the target user will be recommended objects similar to the ones he preferred in the past (see [24, 25] the investigation of OCF algorithms as well as the comparison between UCF and OCF). Using also the Sørensen index, the similarity between two objects, o_i and o_j, can be written as

$$s^o_{ij} = \frac{1}{\sqrt{k(o_i)k(o_j)}} \sum_{l=1}^{m} a_{il} a_{jl}, \tag{8}$$

where the superscript emphasizes that this measure is for object similarity. The predicted score, to what extent u_i likes o_j, is given as

$$v_{ij} = \frac{\sum_{l=1,l\neq i}^{n} s^o_{jl} a_{li}}{\sum_{l=1,l\neq i}^{n} s^o_{jl}}. \tag{9}$$

To test the algorithmic accuracy, we use two benchmark data sets, namely *MovieLens* (http://www.grouplens.org/) and *Netflix* (http://www.netflixprize.com/). The *MovieLens* data consist of 1682 movies (objects) and 943 users, and users vote for movies using discrete ratings 1–5. We therefore applied a coarse-graining method [16, 18]: a movie has been collected by a user if and only if the giving rating is at least 3 (i.e. the user at least likes this movie). The original data contain 10^5 ratings, 85.52% of which are $\geqslant$3, thus after coarse gaining the data contain 82 520 user–object pairs. The *Netflix* data are a random sampling of all the records of user activities in *Netflix.com*, consisting of 10 000 users, 6000 movies and 824 802 links. Similar to the MovieLens data, only the links with ratings no less than 3 are kept. To test the recommendation algorithms, the data set is randomly divided into two parts: The training set

contains 90% of the data, and the remaining 10% of data constitutes the probe. The training set is treated as known information, while no information in the probe set is allowed to be used for recommendation.

A recommendation algorithm should provide each user with an ordered queue of all its uncollected objects. For an arbitrary target user u_i, if the relation $u_i - o_j$ is in the probe set (accordingly, in the training set, o_j is an uncollected object for u_i), we measure the position of o_j in the ordered queue. For example, if there are 1000 uncollected movies for u_i, and o_j is 10th from the top, we say the position of o_j is 10/1000, denoted by $r_{ij} = 0.01$. Since the probe entries are actually collected by users, a good algorithm is expected to give high recommendations for them, thus leading to small r. Therefore, the mean value of the position value $\langle r \rangle$, called *ranking score*, averaged over all the entries in the probe, can be used to evaluate the algorithmic accuracy: the smaller the ranking score, the higher the algorithmic accuracy and vice versa. Note that the number of objects recommended to a user is often limited, and even given a long recommendation list, real users usually consider only the top part of it. Therefore, we adopt in this paper another accuracy index, namely *precision*. For an arbitrary target user u_i, the precision of u_i, $P_i(L)$, is defined as the ratio of the number of u_i's removed links (i.e. the objects collected by u_i in the probe), $R_i(L)$, contained in the top-L recommendations to L, say

$$P_i(L) = R_i(L)/L. \tag{10}$$

The precision of the whole system is the average of individual precisions over all users, given as

$$P(L) = \frac{1}{m}\sum_{i=1}^{m} P_i(L). \tag{11}$$

Since the ranking score does not depend on the length of recommendation list, hereinafter, unless stated otherwise, the optimal value of a parameter is always subject to the lowest ranking score. In tables 1 and 2, we report the algorithmic performance for *MovieLens* and *Netflix*, respectively. Taking into account only the recommendation accuracy, NBI performs better than GRM and CF (NBI performs remarkably better than UCF, better than OCF for ranking score and competitively to OCF for precision).

3. Improved algorithm by eliminating redundant correlations

In NBI, for any user u_i, the recommendation value of an uncollected object o_j is contributed by all u_i's collected objects, as

$$f'_j = \sum_l w_{jl} a_{li}. \tag{12}$$

Those contributions, $w_{jl}a_{li}$, may result from similarities in the same attributes, thus leading to heavy redundancy. We use an illustration, as shown in figure 2, to clarify our idea. Here, we assume that all the objects can be fully described by two attributes, color and shape, and the target user, say u_i, likes black and square. In figure 2(a), A and B are collected objects and C is uncollected, while in figure 2(b), D and E are collected and F is uncollected. All five links, representing correlations between objects, should have more or less the same weight in the object–object network since each of them results from one common attribute as labeled. Here, the weight of each link is set to be a unit.

Table 1. Algorithmic performance for *MovieLens* data. The precision, intra-similarity, diversity and popularity are corresponding to $L = 50$. Heter-NBI is an abbreviation of NBI with heterogeneous initial resource distribution, proposed in [18]. RE-NBI is an abbreviation of redundant-eliminated NBI, the algorithm presented in this paper. The parameters in Heter-NBI and RE-NBI are set as the ones corresponding to the lowest ranking scores (for Heter-NBI [18], $\beta_{\mathtt{opt}} = -0.80$; for RE-NBI, $a_{\mathtt{opt}} = -0.75$). Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and probe.

Algorithms	Ranking score	Precision	Intra-similarity	Hamming distance	Popularity
GRM	0.140	0.054	0.408	0.398	259
UCF	0.127	0.065	0.395	0.549	246
OCF	0.111	0.070	0.412	0.669	214
NBI	0.106	0.071	0.355	0.617	233
Heter-NBI	0.101	0.073	0.341	0.682	220
RE-NBI	0.082	0.085	0.326	0.788	189

Table 2. Algorithmic performance for *Netflix* data. The precision, intra-similarity, Hamming distance and popularity are corresponding to $L = 50$. The parameters in Heter-NBI and RE-NBI are set as the ones corresponding to the lowest ranking scores (for Heter-NBI [18], $\beta_{\mathtt{opt}} = -0.71$; for RE-NBI, $a_{\mathtt{opt}} = -0.75$). Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and probe.

Algorithms	Ranking score	Precision	Intra-similarity	Hamming distance	Popularity
GRM	0.068	0.037	0.391	0.187	2612
UCF	0.058	0.048	0.372	0.405	2381
OCF	0.053	0.052	0.372	0.551	2065
NBI	0.050	0.050	0.366	0.424	2366
Heter-NBI	0.047	0.051	0.341	0.545	2197
RE-NBI	0.039	0.062	0.336	0.629	2063

For both C and F, the final recommendation value is two. However, according to our assumption, the target user likes C more than F. This is because in figure 2(a), the recommendations from A and B are independent, resulting from two different attributes; while in figure 2(b), the recommendations resulting from the same attribute (i.e. color = black) are counted twice. Indeed, when calculating the recommendation value of F, the correlations D–F and E–F are redundant for each other. Although real recommendation systems are much more complicated than the simple example shown in figure 2, and no clear classification of objects' attributes as well as no accurately quantitative measurements of users' tastes can be extracted, we believe the redundancy of correlations is ubiquitous in those systems, which depresses the accuracy of NBI.

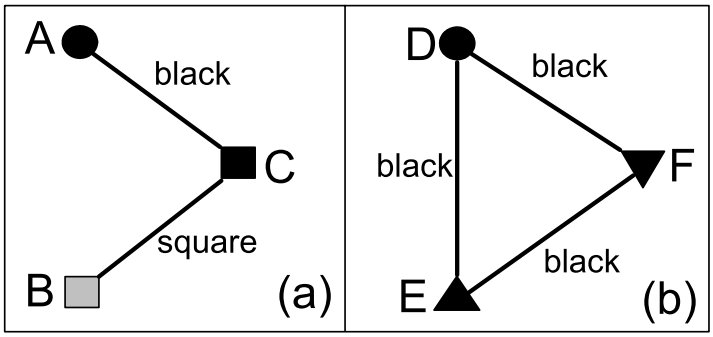

Figure 2. Illustration of redundant correlations.

Note that, in figure 2(a), A and B, sharing no common property, do not have any correlation (in a real system, two objects, even without any common/similar property, may have a certain weak correlation induced by occasional collections). In figure 2(b), D and E are tightly connected for their common attribute, color = black, which is also the cause of redundant recommendations to F. Therefore, following the path D→E→F, D and F have strong second-order correlation. However, since the correlation between A and B is very weak, the second-order correlation between A and C, contributed by the path A→B→C, should be negligible.

Generally speaking, if the correlation between o_i and o_k and the correlation between o_j and o_k contain some redundancy to each other, then the second-order correlation between o_i and o_k, as well as that between o_j and o_k should be strong. Accordingly, subtracting the higher order correlations in an appropriate way could, perhaps, further improve the algorithmic accuracy. Motivated by this idea, we replace equation (5) by

$$\mathbf{f}' = (W + aW^2)\mathbf{f}, \tag{13}$$

where a is a free parameter. When $a = 0$, it degenerates to standard NBI discussed in the previous section. If the present analysis is reasonable, an algorithm with a certain negative a could outperform the case with $a = 0$.

Figure 3 reports the algorithmic accuracy, measured by the ranking score, as a function of a, which has a clear minimum around $a = -0.75$ for both *MovieLens* and *Netflix*. Compared with the standard case (i.e. $a = 0$), the ranking score can be further reduced by 23% for *MovieLens* and 22% for *Netflix*. This result strongly supports our analysis. It is worth emphasizing that more than 20% is indeed a great improvement for recommendation algorithms. In addition, we compare the present algorithm with the *Latent Dirichlet Allocation* (LDA) algorithm [26], which is widely accepted as one of the most accurate personalized recommendation algorithms. Although LDA requires much more computational time, the ranking score for *MovieLens* data is about 0.088, remarkably larger than the minimum, 0.082, obtained by the present algorithm. The accuracy of the present method, even far beyond our expectation, indicates a great significance in potential applications. In figures 4 and 5, we show how the parameter a affects the precision for some typical lengths of recommendation list. Although the optimal value of a leading to the highest precision is different from that subject to the lowest ranking score, the qualitative behaviors of $\langle r \rangle$ versus a and $P(L)$ versus a are the same, that is, in each case, there exists a certain negative a corresponding to the most accurate recommendations (subject to the specific accuracy metric) with remarkable improvement compared with standard NBI at $a = 0$. We compare the ranking score and precision for $L = 50$ in tables 1 and 2, where Heter-NBI

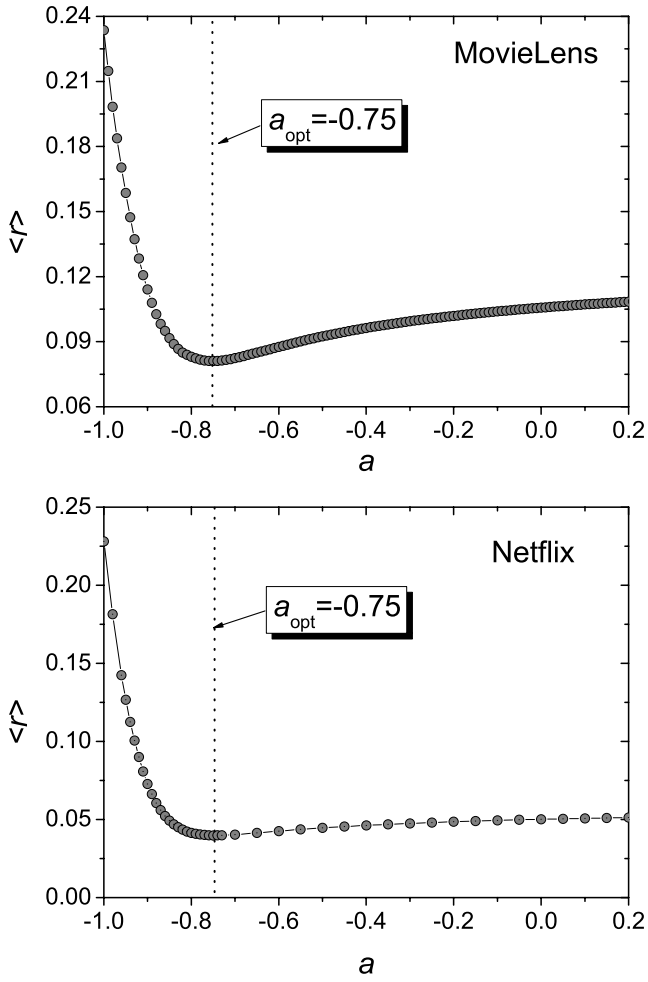

Figure 3. The ranking score $\langle r \rangle$ versus a. The upper and lower plots show the numerical results for *MovieLens* and *Netflix*, respectively. Each data point is obtained by averaging over five runs, each of which has an independently random division of training set and probe. Interestingly, for both *MovieLens* and *Netflix*, the optimal a, corresponding to the minimal $\langle r \rangle$, is $a_{\mathtt{opt}} \approx -0.75$.

represents an improved NBI algorithm with heterogeneous initial resource distribution [18], and RE-NBI is the current algorithm. For fair comparison with parameter-free algorithms, in both Heter-NBI and RE-NBI, the parameters are fixed as those corresponding to the lowest ranking score; therefore the precisions presented in tables 1 and 2 are smaller than the optima. Even so, the present algorithm gives much more accurate recommendations than all the others.

Although without a clear physical picture, equation (13) can be naturally extended to a formula containing even higher order of correlations than W^2, such as

$$\mathbf{f}' = (W + aW^2 + bW^3)\mathbf{f}, \tag{14}$$

where b is also a free parameter. Since the computational complexity increases quickly with the increase of the highest order of W, one should check very carefully if this kind of extension is valuable.

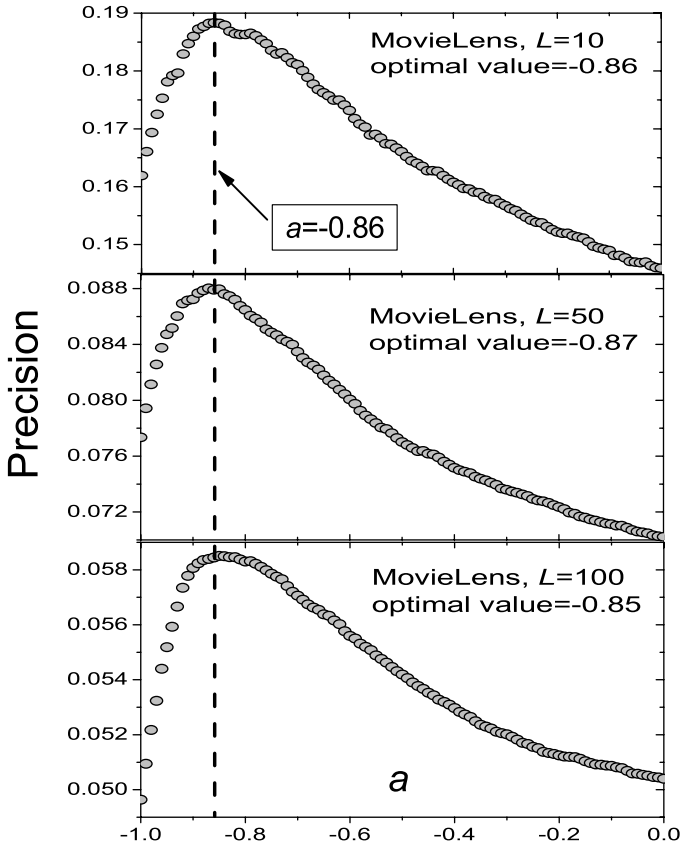

Figure 4. The precision versus a on *MovieLens* data for some typical lengths of recommendation list. Each data point is obtained by averaging over five runs, each of which has an independently random division of training set and probe.

Extensive numerical simulations have been generated to search the global minimum of $\langle r \rangle$ in (a, b) plane for *MovieLens* data. Given a, denoting $b^*(a)$ the optimal value of b corresponding to the smallest $\langle r \rangle$, as shown in figure 6 (red thick line), $b^*(a)$ decreases with the increasing of a in an approximately linear way. The global minimum of $\langle r \rangle$ is about 0.0794, corresponding to $(a^*, b^*) = (-1.6, 0.8)$. That is to say, taking into account the cube of W, the algorithmic accuracy can be further improved by about 3%. However, readers should be warned that the optimal parameters, a^* and b^*, may be widely different for different systems, and finding them will take a very long time for huge systems. Therefore, an algorithm concerning three or even higher orders of the weighted matrix may be not applicable in real systems.

Instead of the global search in (a, b) plane, a possible way to quickly find a nearly minimal $\langle r \rangle$ is to use a greedy algorithm containing two steps. First, we search the optimal a considering only the square of W, as shown in equation (13). Then, we search the optimal b with a fixed as the optimal value obtained in the first step. Clearly, this greedy method runs much faster than the blinding search in the (a, b) plane. However, as shown in figure 7, for *MovieLens* data with $a_{\mathtt{opt}} = -0.75$, the optimal b is zero, giving no improvement of the algorithm shown in equation (13). Therefore, though the introduction of two-order correlation can greatly improve the algorithmic accuracy, to consider three or even higher orders of W may be not valuable.

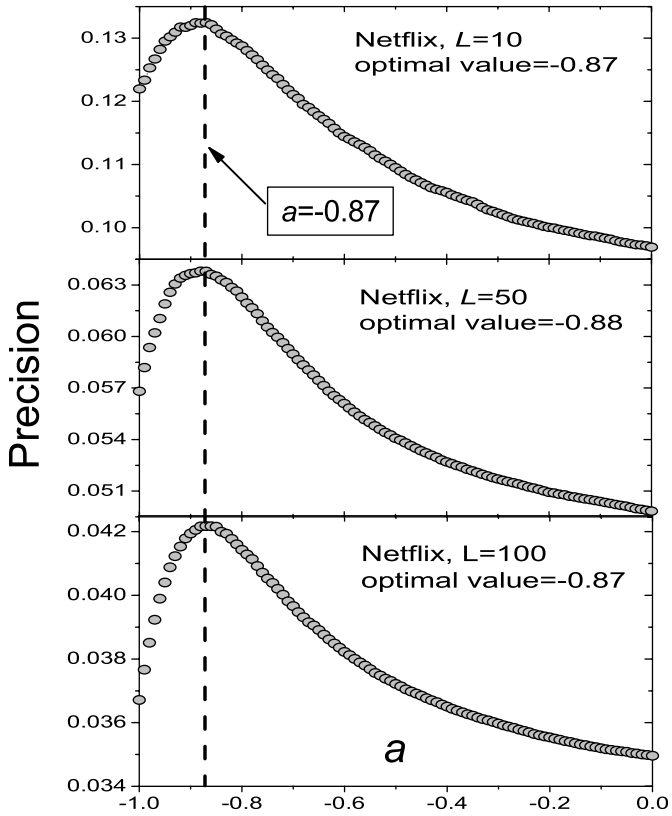

Figure 5. The precision versus a on *Netflix* data for some typical lengths of recommendation list. Each data point is obtained by averaging over five runs, each of which has an independently random division of training set and probe.

4. Popularity and diversity of recommendations

When judging algorithmic performance, most previous works only consider the accuracy of recommendations. Those measurements include [9, 10, 16, 27] *ranking score*, *hitting rate, precision, recall, F-measure* and so on. However, besides accuracy, two significant ingredients must be taken into account. Firstly, the algorithm should guarantee the diversity of recommendations, namely, different users should be recommended different objects. This is also the soul of personalized recommendations. The inter-diversity can be quantified via the *Hamming distance* [18]. Denoting by L the length of recommendation list (i.e. the number of objects recommended to each user), if the overlapped number of objects in u_i and u_j's recommendation lists is Q, their Hamming distance is defined as

$$H_{ij} = 1 - Q/L. \tag{15}$$

Generally speaking, a more personalized recommendation list should have larger Hamming distances than other lists. Accordingly, we use the mean value of Hamming distance,

$$S = \frac{1}{m(m-1)} \sum_{i \neq j} H_{ij}, \tag{16}$$

averaged over all the user–user pairs, to measure the diversity of recommendations. Note that S only takes into account the diversity among users. A good algorithm should also make the

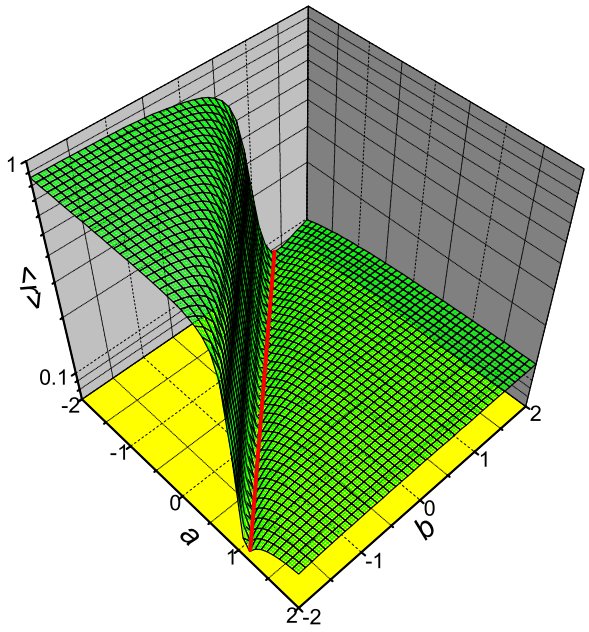

Figure 6. The ranking score $\langle r \rangle$ in (a, b) plane for *MovieLens* data. The numerical simulations run over the parameters, a and b, in the interval $[-2, 2]$ and $[-2, 2]$, respectively, with step length equaling 0.1. To clarify the figure, the axis of $\langle r \rangle$ is set to be logarithmic. Given a, denoting $b^*(a)$ the optimal value of b corresponding to the smallest $\langle r \rangle$. The thick red line emphasizes approximately the function $b^*(a)$. All the numerical results are obtained by averaging over five independent runs with data division identical to the case shown in figure 3. The global minimum is $\langle r \rangle \approx 0.0794$, corresponding to $(a^*, b^*) = (-1.6, 0.8)$.

recommendations to a single user diverse to some extent [28], otherwise users may get tired of receiving many recommended objects under the same topic. Motivated by Ziegler *et al* [28], for an arbitrary target user u_l, denoting the recommended objects for u_l as $\{o_1, o_2, \ldots, o_L\}$, the *intra-similarity* of u_l's recommendation list can be defined as

$$I_l = \frac{1}{L(L-1)} \sum_{i \neq j} s^o_{ij}, \tag{17}$$

where s^o_{ij} is the similarity between objects o_i and o_j, as shown in equation (8). The intra-similarity of the whole system is thus defined as

$$I = \frac{1}{m} \sum_{l=1}^{m} I_l. \tag{18}$$

In this paper, we use S and I, respectively, to quantify the diversities among recommendation lists and inside a recommendation list.

Secondly, with more or less the same accuracy, an algorithm that recommends less popular objects is better than one recommending popular objects. Taking recommendation systems for movies as an example, since there are countless channels to obtain information of popular movies (TV, the Internet, newspapers, radio, etc), uncovering a very specific preference,

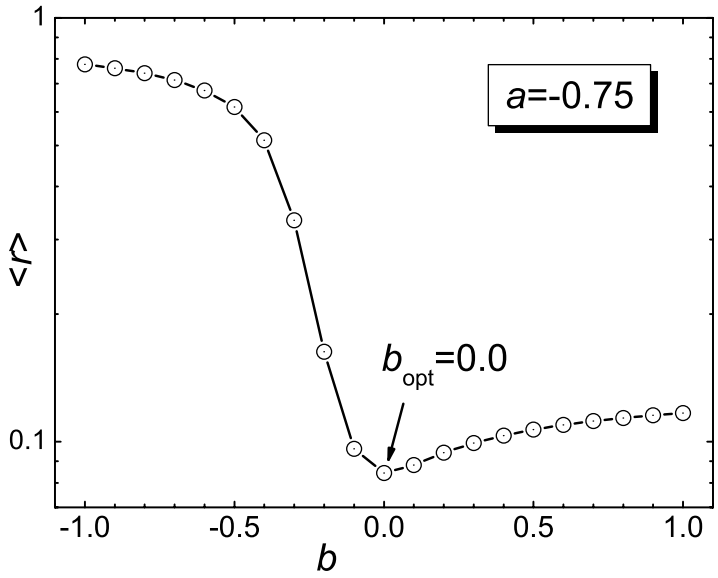

Figure 7. The ranking score $\langle r \rangle$ versus b given $a = -0.75$ for *MovieLens* data. All the numerical results are obtained by averaging over five independent runs with data division identical to the case shown in figure 3. The optimal value of b is zero.

corresponding to unpopular objects, is much more significant than simply picking out what a user likes from the top-viewed movies. The popularity can be directly measured by the average degree $\langle k \rangle$ over all the recommended objects.

Statistically speaking, recommendations displaying high inter-diversity (i.e. large S) will have small popularity. This is because those high-degree (i.e. popular) objects are always the minority in a real system, and highly diverse recommendation lists must involve many less popular objects, thus depressing the average degree $\langle k \rangle$. In contrast, a smaller $\langle k \rangle$ does not guarantee a higher S. An extreme example is to recommend every user the uncollected objects with minimal degree. Therefore the average degree reaches its minimum, while the Hamming distance is close to zero since the recommendations to every user are almost the same. Therefore, S of recommendations provides more information for the algorithmic performance than $\langle k \rangle$. However, the calculation of S takes much longer than that of $\langle k \rangle$, especially for a system containing quite a number of users. In addition, the definition of popularity is simpler and more intuitive than Hamming distance. In comparison, the intra-similarity, I, mainly concerning the underlying content of objects (two objects with similar content or in the same category usually have high probability to be collected by same users), is not directly relevant to the popularity. Therefore, we use all three metrics here to provide a comprehensive evaluation. In a word, besides the accuracy, an algorithm giving higher S, lower I and lower $\langle k \rangle$ is more favorable.

In figure 8, we report the numerical results on how the parameter a affects the Hamming distance, S. From this figure, one can see that the behaviors of $S(a)$ for both *MovieLens* and *Netflix*, as well as for different L, are qualitatively the same, namely S is negatively correlated with a: the smaller a the higher S. As a result, the present algorithm with $a = -0.75$ can provide obviously higher inter-diverse recommendations compared with standard NBI at $a = 0$. Figures 9 and 10 show how the parameter a affects the intra-similarity I and the

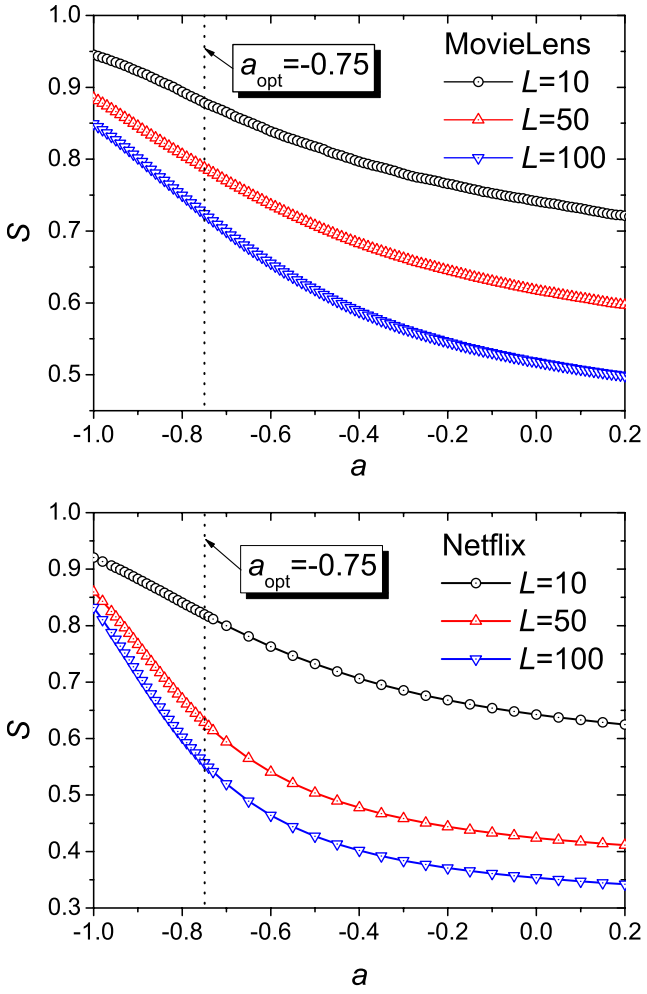

Figure 8. The Hamming distance, S, as a function of a. The black circles, red up-triangles and blue down-triangles represent the cases with typical lengths $L = 10$, 50 and 100, respectively. The upper and lower plots correspond to the results on *MovieLens* and *Netflix*, respectively. The vertical line marks the optimal value of a, as $a_{\mathtt{opt}} = -0.75$. All the numerical results are obtained by averaging over five independent runs with data division identical to the case shown in figure 3.

popularity $\langle k \rangle$, respectively. Clearly, the smaller a leads to less intra-similarity and popularity, and thus the present algorithm can find its advantage in recommending less popular objects with diverse topics to users, compared with standard NBI. Generally speaking, the popular objects must have some attributes fitting the tastes of the majority of the people. Standard NBI may repeatedly count those attributes and thus give overstrong recommendations for the popular objects, which increases the average degree of recommendations, as well as reducing the diversity. CF, considering only the first-order correlations, has the same problem as standard NBI. The present algorithm with negative a can to some extent eliminate the redundant correlations, namely assign lower weights to the most-liked attributes, and thus give higher chances to less popular objects and those objects with diverse topics different from the mainstream.

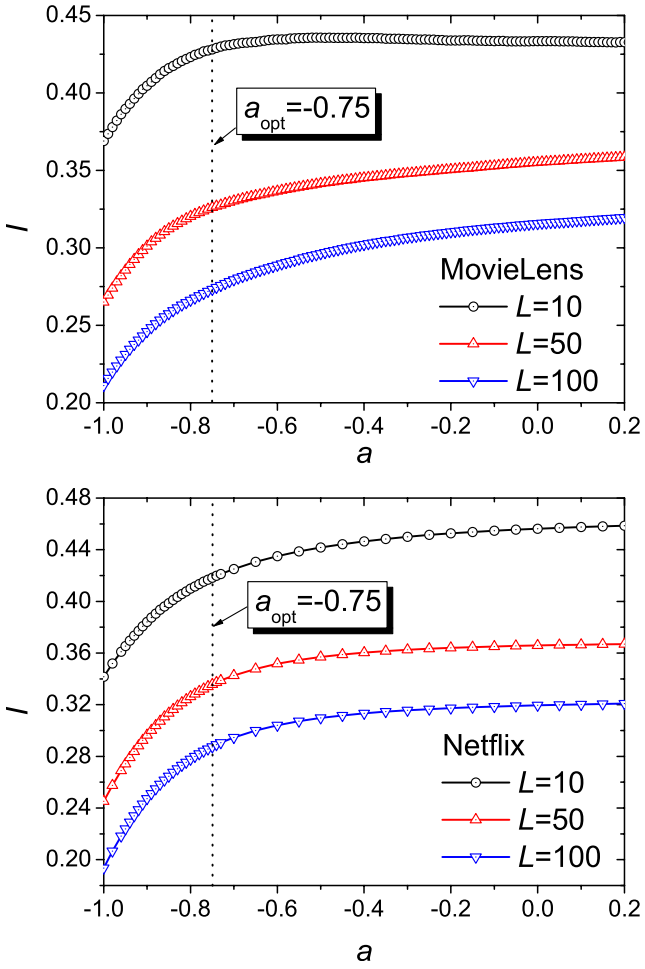

Figure 9. The intra-similarity, I, as a function of a. The black circles, red up-triangles and blue down-triangles represent the cases with typical lengths $L = 10$, 50 and 100, respectively. The upper and lower plots correspond to the results on *MovieLens* and *Netflix*, respectively. The vertical line marks the optimal value of a, as $a_{\mathtt{opt}} = -0.75$. All the numerical results are obtained by averaging over five independent runs with data division identical to the case shown in figure 3.

We summarize the algorithmic performance in tables 1 and 2. One finds that in the case $a = -0.75$, the present algorithm outperforms standard NBI (i.e. $a = 0$) [16] and its variant with heterogeneous initial resource distribution (Heter-NBI) [18] in all five criteria: lower ranking score, higher precision, larger Hamming distance, lower intra-similarity and smaller average degree.

5. Conclusion and discussion

NBI [16], as introduced in section 2, has higher accuracy as well as lower computational complexity than the widely applied personalized recommendation algorithm, namely user-based

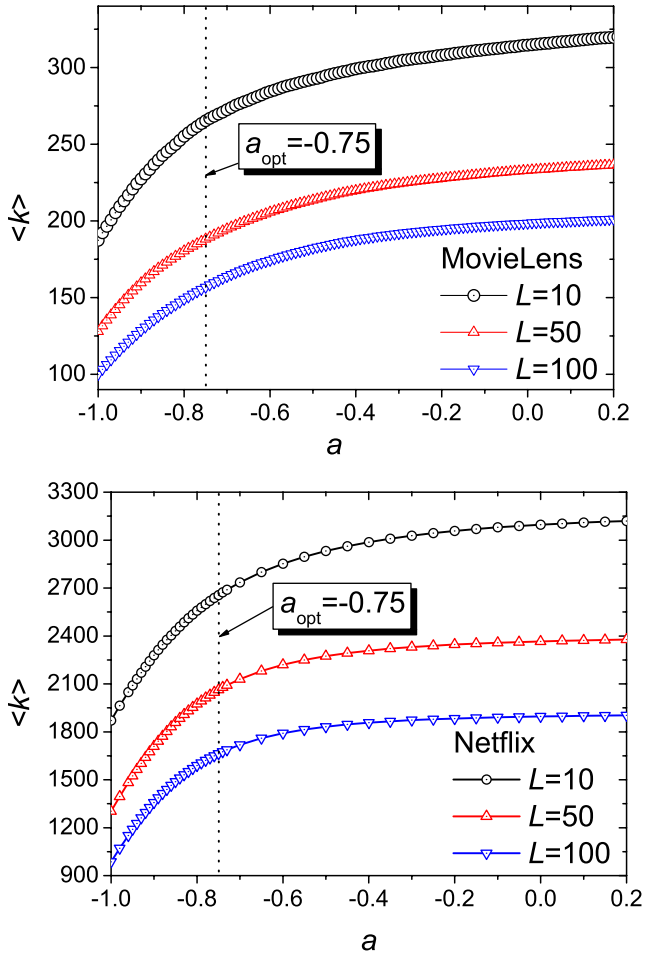

Figure 10. The average degree, $\langle k \rangle$, as a function of a. The black circles, red up-triangles and blue down-triangles represent the cases with typical lengths $L = 10$, 50 and 100, respectively. The upper and lower plots correspond to the results on *MovieLens* and *Netflix*, respectively. The vertical line marks the optimal value of a, as $a_{\mathtt{opt}} = -0.75$. All the numerical results are obtained by averaging over five independent runs with data division identical to the case shown in figure 3.

CF. Therefore, it has great potential significance for practical purposes. However, in this paper, we point out that in NBI, the correlations resulting from a specific attribute may be repeatedly counted in the cumulative recommendations from different objects. Those redundant correlations will depress the algorithmic accuracy. By considering the higher order correlations, W^2, we design an effective algorithm that can, to some extent, eliminate the redundant correlations. The algorithmic accuracy, measured by the ranking score, can be further improved by 23% for *MovieLens* data and 22% for *Netflix* data in the optimal case at $a = -0.75$. Since an algorithm considering even higher orders of W takes too long to be applied in real systems, and the improvement is not much, as shown in figures 6 and 7, we suggest taking into account W and W^2 only.

Any research concerning the accuracy of personalized recommendations should mention the competition *Netflix Prize* [19], which has largely affected the study on recommendation systems. This competition not only provides some algorithms and techniques that have practical significance, but also leads to some scientific insights including statistical regularities about the individual ratings, correlations between the movies rated by a user, strong temporal effects on individual ratings, and so on. Instead of an improvement in the quality of individual algorithms, the more significant discovery arising from this competition is the *ensemble idea*, namely how to properly select and organize many (usually hundreds of) individual algorithms to achieve better prediction accuracy. In fact, the winning team, called *BellKor's Pragmatic Chaos*, is a combined team of *BellKor* [29], *Pragmatic Theory* [30] and *BigChaos* [31] (of course, it is not a simple combination but a sophisticated design), and each of them consists of many individual algorithms (also called predictors, models, etc). For example, the Pragmatic Theory solution considered 453 individual algorithms whose prediction accuracies, measured by root mean square deviation (RSME), range from 0.8762 to 1.1271. The problem studied in this paper is relevant but different from the one concerned in Netflix Prize. Here we focus on the simplest information, collected or not, instead of the ratings for Netflix Prize (we call the latter a rating system). In addition, the predictions made for Netflix Prize usually involve much external information, such as the time of ratings and the content of movies, while the present algorithm does not rely on such information. Although it is easy to degenerate the algorithms for Netflix Prize to the algorithms for the current problem by setting a certain threshold (an object is considered to be collected by a user only if the rating is higher than the threshold), those degenerated versions often perform poorly since the original algorithms are carefully designed to make use of the correlations between ratings. For example, many predictors attempt to train a kind of correlation matrix for different ratings which is meaningful for the current algorithms. Instead, we tried to extend the present algorithm to the rating system by (i) regulating the users' ratings to eliminate the personal bias according to [17]; (ii) building a weighted object–object network according to [17]; (iii) calculating the object similarity based on the two-step diffusion similar to equation (13); (iv) adopting a standard CF technique to obtain the predictions; (v) regulating these predicted ratings to add the personal bias [17] (more details about the extended algorithm are ignored since this is not the main focus of this paper). For the full Netflix Prize data, we finally get RSME ≈ 0.9095 (the goal for Netflix Prize RSME ≈ 0.8563). Without the data regulation, the answer is poor, about 0.9633. For comparison, this result lies in the middle of the individual algorithms considered by the Pragmatic Theory team, and is competitive with but slightly weaker than some other advanced algorithms, such as *regularized* SVD (RSME ≈ 0.9070) by Paterek [32], *iterative self-consistent refinement* (RSME ≈ 0.9038) by Ren *et al* [13], *probabilistic matrix factorization* (RSME ≈ 0.8970) by Salakhutdinov and Mnih [33], *scalable CF* (RSME ≈ 0.8939 to RSME ≈ 0.9046) by Takács *et al* [34] and so on. It is worth emphasizing again that this paper focuses on recommendation systems with unitary data, which are more abundant in the web world since only a very tiny fraction of users are willing to provide ratings. The extended algorithm mentioned here is only used for comparison.

Most previous studies considered algorithmic accuracy only. For example, the Netflix Prize [19] challenged researchers to increase the accuracy without any reference to diversity and novelty. In fact, to predict the ratings on popular movies is much easier than to predict those on unpopular movies, but the latter is more useful since to recommend a very famous movie to a user is usually less creditable. Here, we argue that the diversity and popularity, as the significant criteria of algorithmic performance, should also be taken into account. Diversity

is the soul of a personalized recommendation algorithm, that is to say, different users should be recommended, in general, different objects, and for a single user, the objects recommended to him should contain diverse topics. In addition, the recommendations of less popular objects are very significant in the modern information era, since those objects, even if they perfectly match a user's tastes, could never be found by this user himself from countless congeneric objects (e.g. millions of books and billions of webs). Without recommendation algorithms, those much less popular objects resemble dark information for normal users. Therefore, an algorithm that can provide accurate recommendations for less popular objects can be considered as a powerful tool for uncovering dark information. In a word, with more or less the same accuracy, an algorithm giving higher diversity and lower popularity is more favorable, and the numerical results show that the present algorithm can outperform standard NBI and both user- and object-based CF algorithms simultaneously in all five criteria: lower ranking score, higher precision, larger Hamming distance, lower intra-similarity and smaller average degree. Although this issue (diversity and novelty of recommendations) was discussed in a few early works [28, 35], those were based on restrictive features such as content-specific information and object attributes, while the metrics and methods reported in this paper only require unitary data.

How to better provide personalized recommendations is a long-standing challenge in modern information science. Any answer to this question may intensively change our society, economic and lifestyle in the near future. We believe the current work can enlighten readers in this interesting and exciting direction.

Acknowledgments

We acknowledge the *GroupLens Research Group* for *MovieLens* data and the *Netflix Inc.* for *Netflix* data. This work benefited from Matus Medo and Ming-Sheng Shang who tested the present method (an extended and modified version) in a multi-rating recommendation system (based on the *Netflix* data), and Cihang Jin who provided us the ranking score on *MovieLens* data by using the LDA algorithm. This work is partially supported by SBF (Switzerland) for financial support through project C05.0148 (Physics of Risk), the Swiss National Science Foundation (205120-113842 and 200020-121848), the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 213360 (LIQUIDPUB project), and the National Natural Science Foundation of China under grant no. 60744003. BHW acknowledges the 973 Project 2006CB705500. TZ acknowledges the National Natural Science Foundation of China under grant nos 10635040 and 60973069.

References

[1] Zhang G Q, Zhang G Q, Yang Q F, Cheng S Q and Zhou T 2008 *New J. Phys.* **10** 123027
[2] Broder A, Kumar R, Moghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A and Wiener J 2000 *Comput. Netw.* **33** 309
[3] Brin S and Page L 1998 *Comput. Netw. Syst ISDN.* **30** 107
[4] Kleinberg J M 1999 *J. ACM* **46** 604
[5] Linden G, Smith B and York J 2003 *Internet IEEE Comput.* **7** 76
[6] Billsus D, Brunk C A, Evans C, Gladish B and Pazzani M J 2002 *Commun. ACM* **45** 34
[7] Ali K and van Stam W 2004 *Proc. 10th ACM SIGKDD* p 394

[8] Schafer J B, Konstan J A and Riedl J T 2001 *Data Min. Knowl. Disc* **5** 115
[9] Adomavicius G and Tuzhilin A 2005 *IEEE Trans. Knowl. Data Eng.* **17** 734
[10] Herlocker J L, Konstan J A, Terveen K and Riedl J T 2004 *Trans ACM. Inform. Syst.* **22** 5
[11] Pazzani M J and Billsus D 2007 *Lect. Notes Comput. Sci.* **4321** 325
[12] Maslov S and Zhang Y C 2001 *Phys. Rev. Lett.* **87** 248701
[13] Ren J, Zhou T and Zhang Y C 2008 *Europhys. Lett.* **82** 58007
[14] Goldberg K, Roeder T, Gupta D and Perkins C 2001 *Inf. Retr.* **4** 133
[15] Zhang Y C, Blattner M and Yu Y K 2007 *Phys. Rev. Lett.* **99** 154301
[16] Zhou T, Ren J, Medo M and Zhang Y C 2007 *Phys. Rev.* E **76** 046115
[17] Zhang Y C, Medo M, Ren J, Zhou T, Li T and Yang F 2007 *Europhys. Lett.* **80** 68003
[18] Zhou T, Jiang L L, Su R Q and Zhang Y C 2008 *Europhys. Lett.* **81** 58004
[19] Bennett J and Lanning S 2007 *Proc. KDD Cup Workshop* p 3
[20] Ou Q, Jin Y D, Zhou T, Wang B H and Yin B Q 2007 *Phys. Rev.* E **75** 021102
[21] Sørensen T 1948 *Biol. Skr.* **5** 1
[22] Liben-Nowell D and Kleinberg J 2007 *J. Am. Soc. Inf. Sci. Technol.* **58** 1019
[23] Zhou T, Lü L and Zhang Y C 2009 *Eur. Phys. J.* B **71** 623
[24] Sarwar B, Karypis G, Konstan J A and Riedl J T 2001 *Proc. 10th Int. Conf. WWW* p 285
[25] Liu R R, Jia C X, Zhou T, Sun D and Wang B H 2009 *Physica* A **388** 462
[26] Blei D M, Ng A Y and Jordan M I 2003 *J. Mach. Learn. Res.* **3** 993
[27] Huang Z, Chen H and Zeng D 2004 *ACM Trans Inf. Syst.* **22** 116
[28] Ziegler C N, McNee S M, Knostan J A and Lausen G 2005 *Proc. 14th Int. Conf. WWW* p 22
[29] Koren Y 2009 *The BellKor Solution to the Netflix Grand Prize (Report from the Netflix Prize Winners*)
[30] Piotte M and Chabbert M 2009 *The Pragmatic Theory Solution to the Netflix Grand Prize (Report from the Netflix Prize Winners*)
[31] Töscher A and Jahrer M 2009 *The BigChaos Solution to the Netflix Grand Prize (Report from the Netflix Prize Winners*)
[32] Paterek A 2007 *Proc. KDD Cup Workshop* p 39
[33] Salakhutdinov R and Mnih A 2008 Probabilistic matrix factorization *Advances in Neural Information Processing Systems* ed J C Platt, D Koller, Y Singer and Roweis S (Cambridge, MA: MIT Press)
[34] Takács G, Pilászy I, Németh B and Tikk D 2009 *J. Mach. Learn. Res.* **10** 623
[35] Burke R 2002 *User Model. User-Adap. Interact.* **12** 331